

\$39.95
Atari* 400/800
with at least
32K Disk

QN-LINE systems

presents

THE NEXT STEP

by The Logic Smiths

in Atari Graphics*

*indicates Trade Mark of Atari, Inc.

Atari
THE NEXT STEP
User's Guide

Version 1.0
7 September 1981



by **Bob Stewart**
of **The Logic Smiths**

The information in this document is subject to change without notice and should not be construed as a commitment by On-Line Systems or The Logic Smiths. On-Line Systems or The Logic Smiths assume no responsibility for any errors that may appear in this document.

The software described in this document is furnished under copyright protection and may only be used or copied in accordance with the terms of such legal protection.

Copyright (C) 1981, On-Line Systems

THE NEXT STEP is a trade mark of On-Line Systems.

The following are trade marks of Atari, Inc.:

Atari

Atari 400/800 Personal Computer System

Table of Contents

| | | |
|--|---|----|
| 1.0 | Introduction | 3 |
| 2.0 | Graphics Concepts | 4 |
| 2.1 | Graphic Modes | 4 |
| 2.2 | Character Sets | 5 |
| 2.2.1 | Character Tables | 5 |
| 2.2.2 | Multicolor Characters | 6 |
| 2.2.3 | Character Grouping | 7 |
| 3.0 | Package Disk Handling | 7 |
| 3.1 | Utilities | 8 |
| 3.2 | Example Files | 8 |
| 4.0 | PALETTE | 8 |
| 5.0 | THE NEXT STEP | 9 |
| 5.1 | Usage Flow Chart | 9 |
| 5.2 | General Information | 11 |
| 5.2.1 | Menus | 11 |
| 5.2.2 | Exit Warning | 11 |
| 5.2.3 | File Names | 11 |
| 5.2.4 | Device I/O | 12 |
| 5.2.5 | Character Modes | 12 |
| 5.3 | Choosing Your Initial Character Set | 13 |
| 5.4 | The Crossroads | 15 |
| 5.5 | Editing The Character Set | 16 |
| 5.5.1 | Character Display And Selection | 17 |
| 5.5.2 | Character Editing | 18 |
| 5.5.3 | Top Menu Region | 20 |
| 5.6 | Experimental Display | 20 |
| 5.6.1 | Character Selection | 21 |
| 5.6.2 | Building A Display | 21 |
| 5.6.3 | Top Menu Region | 21 |
| 5.7 | Generating Code For Your Program | 22 |
| 5.8 | File Format | 24 |
| 6.0 | Mixed Mode Graphics | 24 |
| 6.1 | Display List | 25 |
| 6.2 | Mixed Mode Output | 27 |
| 7.0 | Programming Examples | 27 |
| 7.1 | Modified Character Set | 28 |
| 7.2 | Multicolor Character Animation | 29 |
| 7.3 | Mixed Mode Graphics | 30 |
| APPENDIX A <i>How to Type or Display any character</i> | | |

Introduction

1.0 Introduction

The Atari 400/800 Personal Computer System is a powerful graphics computer. Unfortunately, as is usually the case with flexible systems, many of the features are rather complicated to understand and use. The Atari *THE NEXT STEP* Graphics Package helps you use some of these features, whether you are a beginning BASIC programmer or an advanced assembly language programmer.

The features addressed by *THE NEXT STEP* package revolve around the Atari concepts of graphic modes and character sets, both of which are discussed in detail in this guide. Briefly, *THE NEXT STEP* package helps you with:

Intermixing of many graphic modes at the same time. For example: normal text, large text, low resolution plotting, high resolution plotting. This allows you to format a much more complex display than BASIC supports.

Defining and using character sets containing characters other than the Atari standard letters, numbers, punctuation and graphic characters. This has simple, obvious uses such as the definition of foreign alphabets or special symbols. It also has more open ended uses in graphics and animation.

The elements of the Atari THE NEXT STEP Graphics Package are:

This guide.

PALETTE, a program to help you choose colors.

THE NEXT STEP, a Character set Editor program.

A subroutine to simplify mixed code graphics.

Example programs to demonstrate advanced graphics.

To use *THE NEXT STEP* Graphics Package, you must have the following minimum Atari system:

Atari 400/800 Personal Computer.

32K of memory.

A disk drive.

A joystick.

This guide assumes that you are already familiar with the use of the above hardware. It also assumes that you are familiar with Atari BASIC programming as described in the Atari 400/800 BASIC Reference

Manual. Although not assumed by this guide, you can find detailed descriptions of all the Atari Personal Computer System functions in the Atari Personal Computer System Operating System User's Manual and the Atari Personal Computer System Hardware Manual, both available from Atari.

2.0 Graphics Concepts

This section discusses the Atari graphics concepts that are central to *THE NEXT STEP* Graphics Package:

Graphic modes — different techniques for displaying information, text and plot.

Character sets — the elements that are displayed in text graphic modes.

2.1 Graphic Modes

The Atari Personal Computer has two broad classes of graphic modes:

Text — the display elements are characters. In BASIC you generally display them with POSITION and PRINT statements. They are BASIC graphic modes 0,1, and 2.

Plot — the display elements are points. In BASIC you generally display them with PLOT and DRAWTO statements. They are BASIC graphic modes 3 through 8.

When you have more than one graphic mode on the screen at a time; this is called mixed mode graphics. BASIC calls graphic modes 1 through 8 split-screen modes, since they can have a BASIC mode 0 section at the bottom. The split-screen modes represent the only mixed code graphics that BASIC supports. The Atari hardware, however, can do much more than that. It's just that Atari wisely chose not to overwhelm the beginner with the true flexibility of the computer.

As usual, the real world is more complex than it appears at first glance. If you look closely at your television screen, you will see that it is made up of many horizontal lines. These are called scan lines. A group of 1 to 16 scan lines make up a graphic mode line. Each graphic mode line is one line of the graphic mode, such as one line of text or one line of plotting. You can mix graphic mode lines any way you want to, up to a total of 192 scan lines (that's all your TV screen can reliably hold).

BASIC supports most of the graphic modes that the Atari hardware supports. However there are a few notable exceptions. There is a mode similar to BASIC mode 0 except that it has deeper descenders on lower case characters such as p and g. There are a couple of text modes that allow multiple colors within the same character (more on this in the next section). And there are a couple of plotting modes that have rectangular points. All of the modes are implemented in the hardware by an Atari custom designed integrated circuit called ANTIC. ANTIC uses mode numbers to identify graphic modes and, unfortunately, they are different from BASIC's.

This guide will therefore always qualify graphic modes as either ANTIC mode or BASIC mode. For example, ANTIC mode 2 is the same as BASIC mode 0. More details on all of this are in the later section on mixed mode graphics.

2.2 Character Sets

This section contains a lot of information about character sets. Some of it is very important. Some of it is of general interest. Eventually, you will probably want to understand all of it and more. For now, read on and don't worry if you don't understand it all.

When you are programming your own special character sets, you will find yourself wanting to type or display characters you normally don't think of in that way. For example, do you know how to type an inverse video up-arrow? Or display one? There is an appendix at the end of this guide with a copy of the character table from the BASIC manual and some additional notes that will help you solve this problem.

2.2.1 Character Tables

The elements you display in the text graphic mode come from a character set. Each character you see on the screen has a character number. The character number is what you get when you use the BASIC ASC function. That character number is used by ANTIC to find a little table that tells it exactly how to display the character. Each character has a table that is 8 bytes long. Just in case you don't know, a byte is a single memory location. It is 8 bits long (a bit is a binary digit, either 0 or 1). That means that each character is completely described by a table containing 64 bits (8 bytes times 8 bits each). The bits in the table tell ANTIC what color to display for the corresponding point in the character. Each point in each character is called a pixel,

for picture element.

The character tables are collected into a big table. This big table describes a character set. The Atari has 128 predefined characters. The table for this character set is in system Read-Only Memory (ROM). ANTIC knows where the character set table is by looking at a character page number that you can control (a page is a set of 256 bytes on an even 256 byte boundary). Because of ANTIC's internal pickiness, the page number must be evenly divisible by 4 in some cases and 2 in others. Fortunately, *THE NEXT STEP* package will take care of most of that for you. You only have to know that you can change which character set is being displayed simply by changing the page number that ANTIC is using.

Character sets come in two sizes, depending on the text graphic mode. BASIC mode 0 uses a table of 1024 bytes (128 characters times 8 bytes each). BASIC modes 1 and 2 use a table of 512 bytes (64 characters times 8 bytes each). The page number for 1024 byte table must be evenly divisible by 4. The page number for a 512 byte table must be evenly divisible by 2. *THE NEXT STEP* package takes care of most of this for you, too. So don't worry.

2.2.2 Multicolor Characters

One of the really nice features that ANTIC supports but BASIC doesn't is multicolor characters. You know you can have four different colors in BASIC modes 1 and 2, but this is a little different. ANTIC has two graphic modes where each pixel in a character can be one of four colors. This means that a character can be displayed in more than just one color and the background. You can have three colors in the same character, plus the background. And, similar to inverse video for normal characters, you can exchange one of those three for a fourth.

Multicolor characters come in short and tall varieties. The short ones are sort of like BASIC mode 1 and the tall ones are sort of like BASIC mode 2. The thing that's a little strange about the multicolor characters is that they are only four pixels wide. In other words they are displayed as half the width of the BASIC versions. That means that it takes two multicolor characters to make up one BASIC style character.

The color of multicolor characters is controlled with SETCOLOR 0, 1, 2, and 3. The colors from 2 and 3 are the ones you can swap for each

other. You control the swap by displaying the character as normal or inverse (inverse is the normal character number plus 128).

This should be enough to get you excited about the fancy graphics you can do (imagine a character that looks like a little sailboat, with 3 colors).

2.2.3 Character Grouping

There is one final bit of complexity that you need to hear about. Deep in its insides, the Atari computer uses different character numbers than you hear about in BASIC's manual. This is for purposes of its own that we won't go into, but you need to hear a little about it. The upshot of the situation is that characters are not arranged in the ROM the same way they are in ATASCII. In the ROM, all the upper case and punctuation characters are in the first 512 bytes and all the lower case and graphics characters are in the second 512 bytes. That's why you only get upper case in BASIC modes 1 and 2. If you want lower case in those modes you have to POKE 226 into location 756, but then you loses the others. That's all because BASIC modes 1 and 2 can only have 64 characters. For the most part you won't need to worry about this, but you need to have the general idea.

By the way, remember the character set table page number that ANTIC uses? In case you hadn't guessed, that's what's at location 756. And 226 is the page number of the second half of the ROM character set table (224 is the first half).

3.0 Package Disk Handling

This section discusses *THE NEXT STEP* package disk. After reading this section you will know what's on *THE NEXT STEP* disk and what you can do with it.

There are two kinds of programs on your *THE NEXT STEP* disk:

Utilities — the PALETTE and THE NEXT STEP programs. These cannot be copied or modified and can only be used as furnished.

Examples — simple programs utilizing modified character sets and mixed mode graphics can be individually copied for your use and experimentation.

3.1 Utilities

To run the utilities, set up your computer with disk drive, BASIC cartridge, and a joystick in the far left position as described in the Atari manuals. Put *THE NEXT STEP* disk in disk drive number 1 and turn on the system. Your computer will boot and run a menu program that lets you select either *PALETTE* or *THE NEXT STEP*. A blinking red asterisk indicates which program is selected. Use the joystick to move the asterisk. When it is on the program you want, press the joystick trigger and the selected program will run. While the utilities are running, the break key has no function and is ignored. If you press *SYSTEM RESET*, the loading process will repeat from the beginning. If any fatal error occurs while loading a utility, the system will attempt to reboot.

When you exit *PALETTE* or *THE NEXT STEP* (as described in their own sections) you will either be returned to the menu, if *THE NEXT STEP* disk is still in place, or the system will attempt to reboot from the disk that is in place.

While you are running *THE NEXT STEP*, you should use one of your own disks to contain the files that *THE NEXT STEP* creates for you.

3.2 Example Files

To run the example programs, you must boot your system with one of your own disks. You can then *LOAD* or *RUN* the programs as you normally would any BASIC program.

The character sets for the example programs are on the disk. To examine or use them, use *THE NEXT STEP* to load the character set, then change to a disk of your own and use *THE NEXT STEP* command to save the character set.

4.0 PALETTE

PALETTE is a relatively simple program that makes it easy to experiment with different color combinations. It helps you to determine what color and luminance values to use in a *SETCOLOR* command to get the color you want.

PALETTE lets you experiment by changing the background (*SETCOLOR 4*) and/or colors 1 and 2 (*SETCOLOR 1* and *2*). As you change color and luminance values the bottom section of the

PALETTE screen shows the effects. The value that you have selected to change is identified by a blinking asterisk. You move the blinking asterisk with the joystick, and change the value with the joystick trigger. As you hold the trigger, the value counts up. When it reaches the maximum allowed, it resets to zero and counts up again.

Once you have determined the colors you like, simply write them down (or remember them) and put them in the SETCOLOR statements in your program.

You leave PALETTE by pressing any of the console keys (START, SELECT or OPTION).

5.0 THE NEXT STEP

THE NEXT STEP is the centerpiece of the package, the character set editor. It is a fairly large program with a lot of options. This section will tell you some general things about *THE NEXT STEP*, then it will tell how to use *THE NEXT STEP* to create and modify your own character sets.

Briefly, *THE NEXT STEP* can do the following:

- Edit character sets of 64 characters (BASIC modes 1 and 2), 128 characters with inverse video (BASIC mode 0), or 128 characters with multicolor pixels (not directly supported by BASIC).

- Start you out with a standard character set copied from the system ROM, a set of all blanks, or a set previously stored on disk or cassette.

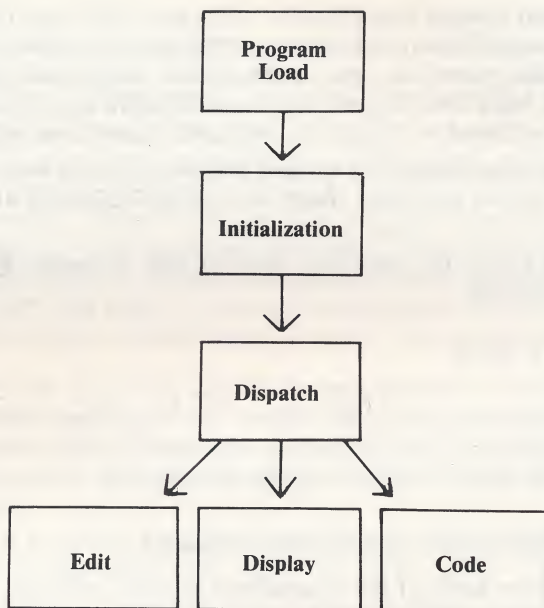
- Store a modified character set on disk or cassette.

- Test arrangements of characters in graphic displays.

- Write BASIC code that makes it easy to integrate your new character set into your program.

5.1 Usage Flow Chart

This section describes the paths that you can take through *THE NEXT STEP* to accomplish your purpose. You can think of these as the steps you take to get where you want to go. First you need *THE NEXT STEP* roadmap:



The boxes name the steps. The arrows indicate the directions you can go. Each step is briefly explained below. They are then explained in detail in the following sections.

Program load — getting *THE NEXT STEP* loaded from *THE NEXT STEP* disk and letting *THE NEXT STEP* set up its internal information. Since this takes a while (*THE NEXT STEP* is rather long) there is a pretty screen for you to look at. For your general information, the screen is composed of BASIC mode 2 and multicolor text, and uses lots of redefined characters.

Initialization — choice of the initial values in the character set that you wish to edit. This is the first time *THE NEXT STEP* lets you choose what to do next.

Dispatch — selection of other steps, plus a few other functions such as saving your character sets. This is the central place to which you return from the other steps.

Edit — editing your character set. This step also displays the character set as edited so far.

Display — arrangement of characters from your character set into experimental displays so that you can see how they look together.

Code — automatic writing of BASIC code that you can ENTER into programs that are to use the edited character set.

5.2 General Information

This section describes general information about *THE NEXT STEP* that applies across more than one step in its use.

5.2.1 Menus

You select *THE NEXT STEP* options from menus using the joystick and trigger. Each menu is of a different screen color and has its name at the top. The menu briefly describes each of its options in a single phrase preceded with an asterisk. There is a blinking asterisk in front of the selected option. Use the joystick to select a different option. Run the selected option by pressing the joystick trigger.

Most of the time if you try to move the asterisk off the menu, it just wraps around to the other side. A few places where this is not true are mentioned in the description of that menu.

5.2.2 Exit Warning

When you select an exit *THE NEXT STEP* option, *THE NEXT STEP* checks to see if you have modified the character set since last saving it. If not, *THE NEXT STEP* simply exits. If so, *THE NEXT STEP* prompts "Exit without saving changes?" on a red screen. If you respond with anything that starts with an upper or lower case Y, *THE NEXT STEP* exits. Any other response from you results in a return to the menu from which you selected the exit option so you can save your work.

5.2.3 File Names

THE NEXT STEP needs names for input of saved character set, for output to save a character set, and for output of code to use a character set. *THE NEXT STEP* allows you to specify a file name when certain menu options are chosen. All file names have the same form.

THE NEXT STEP prompts with a short message that identifies the file name being requested. You then type a file or device name of up to 16 characters, terminated with a carriage return. The usual Atari line

editing functions apply. If you type too many characters, *THE NEXT STEP* prints the error message "File name too long" and prompts again. If you respond with only a carriage return, the state of the file name is left unchanged and *THE NEXT STEP* proceeds according to what you were doing before it prompted you.

If the file name is less than 15 characters long and does not contain a colon (:), *THE NEXT STEP* inserts the characters "D:" at the beginning of the file name for you.

5.2.4 Device I/O

THE NEXT STEP can get device I/O errors while reading or writing a character set or while writing code. They can occur either on attempting to open the file or device or while reading or writing it. In any case, *THE NEXT STEP* will print one of the following error messages. It will then proceed according to what you were doing.

THE NEXT STEP I/O error messages are:

No such file or device — There was some error in the file or device name you supplied

Disk or file write protected — You attempted to write on a protected disk or a locked file.

Disk full — There is no more room left on the disk you specified.

Device I/O error #◀error▶ — The I/O error was not one of the above. ▶error▶ is the Atari standard error number.

5.2.5 Character Modes

THE NEXT STEP supports several different character modes. These modes effect the way that characters are displayed and edited.

THE NEXT STEP treats sets of 64 characters as BASIC Mode 1. *THE NEXT STEP* displays them as grey on a black background.

For sets of 128 characters, *THE NEXT STEP* has 3 options:

Single color — this is BASIC Mode 0. *THE NEXT STEP* displays the characters as grey on a black background and includes the inverse video.

Short multicolor — this is not supported by BASIC. *THE NEXT*

STEP displays the characters as white, grey, and blue on a black background. Possible change in colors.

Tall multicolor — this is not supported by BASIC. The colors are the same as for short multicolor.

THE NEXT STEP does not use the hardware capability to display a fourth color in multicolor characters. In *THE NEXT STEP* displays of multicolor characters. Blue is the color that could be swapped for a fourth.

The colors in *THE NEXT STEP* multicolor character display correspond to SETCOLOR as follows:

| THE NEXT STEP color: | SETCOLOR Number: |
|-------------------------|---------------------|
| Black | 4 (background) |
| White | 0 |
| Grey | 1 |
| Blue | 2 or 3 |

When you edit a 128 character set, *THE NEXT STEP* defaults to single color mode. You can change character modes with a menu option that is described later.

Because of the hardware display characteristics, there is a potentially confusing aspect of multicolor character displays that you must understand. This problem arises in any case where *THE NEXT STEP* screen region contains a normal display of characters from the set being edited.

The only way to have a blank space in these regions is to have a blank character in the character set. If you have a non-blank character instead of the usual blank, that character will be displayed in those parts of the region that would normally be blank. *THE NEXT STEP* will continue to operate correctly, but you may want to keep a blank character in its usual place to avoid an unpleasant or confusing *THE NEXT STEP* display.

5.3 Choosing Your Initial Character Set

In this step you choose the character set you want to start with. You choose from the brown Initialization Menu.

The Initialization Menu options are as follows. Each option explanation is identified by the same line that appears on the screen in actual use.

***Previously saved file**

If you select this option, *THE NEXT STEP* asks for a file name with the prompt "Input file name?". After you supply the required input, *THE NEXT STEP* attempts to read a character set from the file, assuming either a 64 or 128 character set according to the length of the file.

At any time, a carriage return in response to the prompt returns you to the Initialization Menu.

If *THE NEXT STEP* encounters an error opening the file, it prints an error message and reissues the prompt.

If *THE NEXT STEP* encounters any error reading the file, it prints the error message, prompts "Type RETURN to proceed" and waits for you to type on the keyboard, then returns to the Initialization Menu.

The possible error messages for this process are:

No such file or device — There was some error in the file or device name you supplied.

File too short — The file is less than 512 bytes long (that's the minimum length of a character set). You probably gave *THE NEXT STEP* a bad file name, and it tried to load something that wasn't a character set.

Device I/O error #◀error▶ — The I/O error was not one of the above. ◀error▶ is the Atari standard error number.

*** 64 blanks**

If you select this option, *THE NEXT STEP* sets up a 64 character set of all blanks. That way you can start with a clean slate.

*** 64 upper case, numbers & punctuation**

If you select this option, *THE NEXT STEP* sets up a 64 character set containing the standard upper case alphabetics, numbers, and punctuation characters from the first half of the system ROM.

*** 64 lower case & graphics**

If you select this option, *THE NEXT STEP* sets up a 64 character set containing the standard lower case and graphics characters from the last half of the system ROM.

*** 128 blanks**

If you select this option, *THE NEXT STEP* sets up a 128 character set of all blanks. This gives you a bigger clean slate, especially well suited

for defining a whole new set of multicolor characters.

*** 128 full set**

If you select this option, *THE NEXT STEP* sets up a 128 character set containing all the standard characters from system ROM.

*** Exit *THE NEXT STEP***

If you select this option, *THE NEXT STEP* does its standard exit processing as described earlier.

*** *THE NEXT STEP* Main Menu**

This option only appears if you came to the Initialization Menu from the Main Menu. If you select this option, *THE NEXT STEP* simply returns to the Main Menu without re-initializing.

5.4 The Crossroads

At this point in *THE NEXT STEP*, you see the blue-grey Main Menu. The Main Menu is the point to which all the menus return.

The Main Menu options are as follows. Each option explanation is identified by the same line that appears on the screen in actual use.

*** Edit character set**

If you select this option, *THE NEXT STEP* goes to the Edit step.

*** Experimental display**

If you select this option, *THE NEXT STEP* goes to the Display step.

*** *THE NEXT STEP* Code Menu**

If you select this option, *THE NEXT STEP* goes to the Code Menu.

*** *THE NEXT STEP* Initialization Menu**

If you selected this option, *THE NEXT STEP* goes to the Initialization Menu.

*** Exit *THE NEXT STEP***

If you select this option, *THE NEXT STEP* does its standard exit processing, as described earlier.

*** Save Characters on <file>**

If you select this option, *THE NEXT STEP* writes the character set to the file displayed as <file>. Unless you change it (see next option description), <file> is either the default of "D:TEMP.CHA" or the file you specified to load the character set from. The file writing process

works as follows:

THE NEXT STEP displays "Writing file" in the center of the screen. If any errors are encountered, *THE NEXT STEP* displays an I/O error message as described earlier. *THE NEXT STEP* then prompts "Type RETURN to proceed" and waits for the user to type on the keyboard. *THE NEXT STEP* writes a file of 513 or 1025 bytes, for 64 or 128 character sets, respectively. The file format and content is described in the later section on File Format.

When this option is finished, either successfully or with a failure, *THE NEXT STEP* returns to the Main Menu.

*** Change save file name**

If you select this option, *THE NEXT STEP* prompts "Character set output file name?". You can then specify a new output file as described in the earlier section on file names. Once the prompt has been satisfied, *THE NEXT STEP* returns to the Main Menu.

*** Change character mode**

This line only appears when you are editing a 128 character set.

If you select this option, *THE NEXT STEP* goes to the purple character mode sub-menu. Using this sub-menu, you can select one of the character modes explained in the earlier section on General Characteristics (single color, short multicolor, and tall multicolor.).

5.5 Editing The Character Set

In this step you actually change your character set.

The edit step is basically the same regardless of the number of characters (64 or 128) in the character set, or the character mode for 128 character sets (single color, short multicolor, or tall multicolor). The few differences are described below.

The edit screen contains four regions, in the following order from the top of the screen.

Top menu region - a small menu on a blue line, used to leave the edit step.

Character edit region — a display of the character being edited.

Instruction region — on three blue lines, the Edit Menu or instructions to the user depending on what you are doing.

Character set region — the character set as edited so far.

Each of the regions has a cursor of its own, with only one being active at a time. You move the cursors with the joystick. You move between regions by attempting to move a cursor up or down into the next region. The specific characteristics of the regions and their cursor movement are described below.

5.5.1 Character Display and Selection

At the bottom of the screen, the character set region displays the character set that you are editing. This display has a black background with the characters colored according to the character mode as described earlier.

THE NEXT STEP always displays the character set cursor in this region. The character set cursor is a box that outlines one of the characters like a picture frame. In the character identification display, *THE NEXT STEP* describes the character that is in the cursor. The character identification display is on the right hand side of the upper region of the screen that also contains the character edit window. The character edit window is described later.

The character identification display identifies the character inside the character set cursor. The top line of the character identification display is the message "CHAR#". The second line is the decimal character number of the character. The third line is the character that normally goes with that character number. You use either the character number or the normal character to represent the edited character in a PRINT statement.

The character set cursor and the character identification display are always the same color. This color corresponds to what you are doing. When you are choosing a character to edit, the cursor is green. When a character is being edited and is eligible for storing, the cursor is yellow. When you are selecting a new location for a character (an Edit Menu option described later), the cursor is red.

While you are selecting a character to edit, the instruction region contains the message "Choose the character to edit."

To select a character, you move the cursor with the joystick. If you

move the cursor straight out the top of the region, *THE NEXT STEP* moves you to the top menu region. The cursor will not move below the bottom line. If you move the cursor past the right or left edge of the display, *THE NEXT STEP* wraps the cursor to the other side of the same line.

You select the character in the cursor by pressing the joystick trigger. *THE NEXT STEP* then changes the cursor to yellow and displays the selected character in the edit window for editing.

5.5.2 Character Editing

When you select a character to edit, *THE NEXT STEP* displays that character in the character edit region. At this time, *THE NEXT STEP* also displays the Edit Menu in the instruction region.

The character edit window is a blue frame containing, in most cases, eight horizontal and eight vertical pixel spaces. For multicolor characters there are only four pixels on each horizontal line. In short multicolor character mode, these pixels are double width, so that the frame is the same size as for other modes. In tall multicolor character mode, the pixels are normal width, thus the frame is only half as wide as for other modes. The character is displayed in the colors described earlier.

The cursor in the edit window is a small red square inside one of the pixels. You move the cursor with the joystick. If you move the cursor straight up, past the top line, *THE NEXT STEP* switches to the top line menu. If you move the cursor straight down, past the bottom line, *THE NEXT STEP* switches to the Edit Menu. If you move the cursor left or right, past a window side, *THE NEXT STEP* wraps the cursor around within the window to the opposite side of the same line.

To change the color of a pixel, you press the joystick trigger. In this way the pixel can be cycled through its possible colors, depending on the character mode. In other words, a single color pixel alternates between black (the background) and grey. A multicolor pixel cycles black, each of the three colors described earlier, then back to black.

If you press the trigger and move the cursor at the same time, *THE NEXT STEP* will change the pixel you are in then draw a line of that color. It will stop drawing when you release the controls. While drawing a line, *THE NEXT STEP* will wrap around at the top and bottom of the window.

The Edit Menu contains the following options. Each option explanation is identified by the same line that appears on the screen in actual use.

*** Store**

If you select this option, *THE NEXT STEP* stores the edited character into the position indicated by the character set cursor. *THE NEXT STEP* then moves you to the character set region to choose a character to edit.

*** Abort**

If you select this option, *THE NEXT STEP* aborts the character edit without storing the edited character. *THE NEXT STEP* then moves you to the character set region to choose a character to edit.

*** Relocate**

If you select this option, *THE NEXT STEP* changes the character set cursor and character identification display to red, displays the message "Choose the new location" in the instruction region, and moves you to the character set region to choose a new location for the edited character.

You select a new location in the same way as selecting a character to edit (as explained earlier). You can decide not to select a new location by moving the character edit cursor straight out the top of the character set region. Whether or not you select a new location, *THE NEXT STEP* changes the character set cursor and character identification display to yellow and returns to the Edit Menu.

This option is handy for making a copy of a character in another location in the character set.

*** Rotate ◀arrow▶**

If you select this option, *THE NEXT STEP* rotates all the pixels in the character being edited one space in the direction pointed to by the single character ◀arrow▶. Pixels that would move out of the editing window wrap around to the opposite side. The effect is like wrap-around scrolling.

*** Redirect**

If you select this option, *THE NEXT STEP* changes the direction of the arrow in the "Rotate ◀arrow▶" menu line.

*** Compliment**

If you select this option, *THE NEXT STEP* compliments all the pixels in the character being edited. In single color characters, those that are off are turned on and those that are on are turned off. In multicolor characters the effect is interesting, but less usefull.

5.5.3 Top Menu Region

At the top of the screen is a small menu displayed on a single blue line. This menu contains the following options. Each option explanation is identified by the same line that appears on the screen in actual use.

*** Main Menu**

If you select this option, *THE NEXT STEP* returns to the Main Menu.

*** Experimental display**

If you select this option, *THE NEXT STEP* goes to the Display step. To move out of this menu into another edit step region without selecting one of the options, you simply move the menu's blinking asterisk cursor straight down.

5.6 Experimental Display

This step is not necessary in the development of a character set, but it can be useful for seeing if the characters look the way you want them to when arranged into a display. You can use your character set as building blocks to create a picture. Using this step is very similar to using the edit step with respect to general screen layout and coloring. The experimental display screen contains four regions, in the following order from the top of the screen.

Top menu region — a small menu on a blue line, used to leave the experimental display step.

Display region — the character identification display (as described for the edit step) and the experimental display.

Instruction region — on a blue line, instructions to the user.

Character set region — the character set as changed so far (as described for the edit step).

5.6.1 Character Selection

You select a character from the character set region. The character set region is the same as described for the edit step except that there is no case in which the character set cursor is red. When choosing a character for display, the cursor is green. While a character is chosen, the cursor is yellow.

While the user is selecting a character to place in the experimental display, the instruction region contains the message "Choose the character for display".

When the user selects a character, *THE NEXT STEP* changes the instruction region to say "Place the character in the display" and activates the cursor in the display region.

5.6.2 Building a Display

The display region replaces the edit step's character edit region. The display region contains an experimental display and the character identification display. The character identification display is as described for the edit step.

The experimental display is six lines by eight columns. This display has a small red cursor just like the one from the edit window. When you have a character selected for display, you use this cursor to indicate where *THE NEXT STEP* is to display the character. You deposit the selected character in the display by pressing the joystick trigger. When you deposit the character, *THE NEXT STEP* goes back to the character set region to select another, leaving the character set cursor where it was so that the same character can easily be selected again.

You can leave the experimental display region without depositing the character by moving straight out the top for the top menu or straight out the bottom to select a different character.

THE NEXT STEP initializes the experimental display with what is usually the blank character from the character set.

5.6.3 Top Menu Region

At the top of the screen is a small menu displayed on a single blue line. This menu contains the following options. Each option explanation is identified by the same line that appears on the screen in actual use.

* Main Menu

If you select this option, *THE NEXT STEP* returns to the Main Menu.

*Edit Character Set

If you select this option, *THE NEXT STEP* goes to the edit step. To move out of this menu into the display region, simply move the menu's blinking asterisk cursor straight down.

5.7 Generating Code For Your Program

In the code step, you can have *THE NEXT STEP* write code for you. You can then integrate the code into a program that will use the edited character set. *THE NEXT STEP* writes the code in BASIC in a form that can be used with the BASIC ENTER command. You control this step through the green Code Menu.

When writing code, *THE NEXT STEP* uses line number values and increments that you can specify and allows separate line numbers and increments for DATA statements than for subroutine code. *THE NEXT STEP* allows you to specify the output file for the code. *THE NEXT STEP* sets default values during its internal initialization and displays the current values at the top of the Code Menu. You can change the values with one of the Code Menu options discussed below.

When you select one of the code writing options, *THE NEXT STEP* clears the screen, leaving it the same color, and displays "Writing code" in the center. This can take nearly a minute with some options during which your disk may stop whirring, but that's OK, so don't worry.

If at any time in this process *THE NEXT STEP* encounters an error, *THE NEXT STEP* displays an I.O error message as described earlier. *THE NEXT STEP* then prompts "Type RETURN to proceed" and waits for you to press the RETURN key.

When the option is finished, either successfully or with a failure, *THE NEXT STEP* returns to the Code Menu.

The code that *THE NEXT STEP* writes is in the form of BASIC DATA statements and/or a subroutine.

The DATA statements contain the character byte values, one character's worth per line. For the cases where only the non-ROM characters are of interest, the DATA starts with the character number.

The subroutine has REMarks at the beginning to explain its use. When you call it, it moves the alternate character set into properly aligned memory at the end of the existing BASIC program. The subroutine returns the page number of the alternate character set in the variable CHPAGE, but does not instruct the operating system to use it. You must do this with the command POKE 756, CHPAGE.

The Code Menu options are as follows. Each option explanation is identified by the same line that appears on the screen in actual use

*** Load from file**

If you select this option, *THE NEXT STEP* writes code to load the character set from the current character set output file (see section on Main Menu). Note that the code is written to use file number 1, and contains no I/O error control.

*** Load from ROM & DATA**

If you select this option, *THE NEXT STEP* writes code to load the character set from both system ROM and DATA statements. This option is useful for a case then only a few (up to about 10) characters are changed, since it is faster than loading from a file or completely from DATA.

*** Load from DATA**

If you select this option, *THE NEXT STEP* writes code to load the character set from DATA statements. This option results in code that will take longer to load a character set than loading it from a file. Its only advantage is that it makes the program self-contained in one file.

*** DATA for all characters**

If you select this option, *THE NEXT STEP* writes code containing DATA statements for all the characters in the edited character set.

*** DATA for all non-ROM characters**

If you select this option, *THE NEXT STEP* writes code containing only DATA statements for those characters that are different from system ROM.

*** *THE NEXT STEP* Main Menu**

If you select this option, *THE NEXT STEP* returns to the Main Menu.

*** Change defaults**

If you select this option, *THE NEXT STEP* prints current code writing line number, increment, and output file values, and allows you the option of changing any or all of them.

THE NEXT STEP prompts for the values one at a time. If you respond with a carriage return, the value is left unchanged. If you specify a number that is out of range, *THE NEXT STEP* prints "Value out of range" and re-issues the prompt. For any other invalid numeric response, *THE NEXT STEP* prints "Invalid number" and re-issues the prompt. *THE NEXT STEP* prompts for the file name with "Code output file?" and accepts it as explained earlier for file names.

When you have responded appropriately to all the prompts, *THE NEXT STEP* returns to the Code Menu.

5.8 File Format

The only time this might concern you is if you have a character set saved by some other character set edition. If it uses a similar format, the files are compatible. If it does not, you can possibly write a program to convert the file.

THE NEXT STEP file format is quite simple, being mainly a direct binary copy of the character set data as kept in memory. This consists of either 512 or 1024 bytes, for 64 and 128 character sets, respectively.

THE NEXT STEP puts one additional byte at the end of the file. For 64 character sets, this byte records which half of the system character ROM area the character set came from. *THE NEXT STEP* uses this information in some of the code writing options to determine which characters are different from the ROM. For 128 character sets, this byte indicates the last character mode you used for the set (normal, or tall or short multicolor).

Anytime that *THE NEXT STEP* is unable to obtain extra byte, *THE NEXT STEP* assumes the first half of the ROM area as the original set or normal character mode.

6.0 Mixed Mode Graphics

This section describes how you mix graphic modes in ways that BASIC does not directly support. It is more complex than using the BASIC features directly, but the functions are well within the reach of a BASIC program. This section will give you a general idea of how to

deal with these features. You will find understanding this easier in the context of the examples in a later section.

6.1 Display List

As you recall from an earlier section, ANTIC is the hardware component that controls graphics from within your Atari Personal Computer. ANTIC supports graphic modes that BASIC does not.

To mix modes in your display, you must modify the list of mode lines that ANTIC uses. This list is called the display list. The values in it control the operation of ANTIC. To change the graphic mode of a particular line, you simply replace the value in ANTIC's current list with a different one.

Each **graphic mode** also needs some number of bytes to describe what is displayed on the screen for that line. The display list also defines where in your memory ANTIC can find the data that defines the screen contents. We will only consider this in passing, since it is only necessary for very complex displays. Your major concern in this area is that there is enough space for your display list and screen.

The way to accomplish this is to figure out what mode lines you want. The number of mode lines controls the length of the display list. The type of mode line determines the amount of screen memory and the number of scan lines. To fit correctly on any TV screen, your display list must add up to 192 or fewer scan lines. For the easiest use, your display list and screen memory must be shorter than one of the BASIC modes.

You set up your initial display list with a BASIC GRAPHICS statement, then POKE into it the ANTIC mode numbers of the mode lines you want. This must be finished with a special sequence to get ANTIC back to the top of the list. The best way to understand this is to have you refer to the examples rather than try to explain it here.

All of the Atari graphic modes are as follows:

| BASIC Mode | ANTIC Mode | Scan Lines | Screen Bytes | Description |
|---------------|---------------|---------------|-----------------|--------------------------------------|
| . | 0 | 1-8 | 0 | Blank |
| 0 | 2 | 8 | 40 | Normal text |
| . | 3 | 10 | 40 | Normal text with longer descenders |
| . | 4 | 8 | 40 | Short multicolor |
| . | 5 | 16 | 40 | Tall multicolor |
| 1 | 6 | 8 | 20 | Wide text |
| 2 | 7 | 16 | 20 | Big text |
| 3 | 8 | 8 | 10 | Low resolution plot |
| 4 | 9 | 4 | 10 | Medium resolution plot, 1 color |
| 5 | 10 | 4 | 20 | Medium resolution plot, 3 colors |
| 6 | 11 | 2 | 20 | High resolution plot, 1 color |
| . | 12 | 1 | 20 | High resolution short plot, 1 color |
| 7 | 13 | 2 | 40 | High resolution plot, 3 colors |
| . | 14 | 1 | 40 | High resolution short plot, 3 colors |
| 8 | 15 | 1 | 40 | Super resolution plot |

The memory used by each BASIC graphics mode is as follows. The numbers include space that BASIC may not use but you may for your mixed mode graphics.

| BASIC Mode | Display List | Screen Memory |
|---------------|-----------------|------------------|
| 0 | 32 | 960 |
| 1 | 34 | 640 |
| 1 + 16 | 32 | 640 |
| 2 | 24 | 400 |
| 2 + 16 | 20 | 400 |
| 3 | 34 | 400 |
| 3 + 16 | 32 | 400 |
| 4 | 54 | 640 |
| 4 + 16 | 56 | 640 |
| 5 | 54 | 1120 |
| 5 + 16 | 56 | 1120 |
| 6 | 94 | 2080 |
| 6 + 16 | 104 | 2080 |
| 7 | 190 | 4000 |
| 7 + 16 | 200 | 4000 |
| 8 | 256 | 7840 |
| 8 + 16 | 282 | 7840 |

6.2 Mixed Mode Output

The disadvantage of changing the display list is that BASIC's assumptions about how to put your data into screen memory no longer work. Not to worry! Your *THE NEXT STEP* disk contains a subroutine that neatly solves the problem. To add the subroutine to your program simply ENTER "D:FIXPOS.LIS". Don't worry if some of it looks a little strange, the real working part is written in assembly language. All you have to know is how to use it to fake out BASIC.

The problem that we have is that BASIC thinks it knows how far into screen memory to go to find the position you ask for with a POSITION or PLOT statement. When you change the display list, this no longer works. The FIXPOS subroutine convinces BASIC that screen memory starts somewhere other than where BASIC thought. It also makes BASIC think you are using a different graphic mode than BASIC assumed from your GRAPHICS statement.

The way you use FIXPOS is to initialize it with a GOSUB 20000. That sets up FIXPOS itself. Then, when you want to use one of your mixed mode areas, you use FIXPOS to tell BASIC that that area is the top of the screen. That means after you call FIXPOS, you can use the BASIC I/O statements as usual, as if the beginning of the mode line you told FIXPOS is the upper left hand corner of the screen. In text modes you have to use a POSITION statement get BASIC to the beginning after you call FIXPOS. Again, it will all seem simpler when you study the examples.

The only warning is that by using FIXPOS and a mixed mode screen, you are defeating all the "nice cursor out of range" help that BASIC gives you and you can end up with some pretty weird displays when you have bugs. Fortunately, you can always get back to normal by pressing SYSTEM RESET.

Another thing you need to be aware of is that PRINT statements have more than one format. Remember, you use a simple PRINT in GRAPHICS 0, but a PRINT #6 in GRAPHICS 1 or 2. You also use a simple PRINT for multicolor characters.

7.0 Programming Examples

This section gives you examples of how to use modified character sets and mixed mode graphics. It contains the following examples:

Simple use of a character set with just a few changes.

Animation with multicolor characters, including a simple display list modification.

A display with a few modified characters and an extensively changed display list.

7.1 Modified Character Set

In this example, three of the characters in the set of 64 upper case, numbers and punctuation have been changed to make an anvil when printed side by side. The program modifies the character set with code written by *THE NEXT STEP* then displays four anvils, one in each of the default colors for BASIC mode 2.

This program is on your *THE NEXT STEP* disk as ANVIL. Its character set is there, too, as ANVIL.CHA.

Line 10 calls *THE NEXT STEP* written subroutine that copies the ROM character set and changes the three modified characters from DATA statements.

Line 20 sets a regular BASIC graphics mode.

Line 30 changes to the modified character set. Try removing it to see what happens.

Line 40 positions the cursor for the first two anvils and line 50 displays them. The characters ◀, =, and ▶ are the ones that were replaced with anvil pieces. All other characters are intact. The normal and inverse versions of these characters each get a different color.

Lines 60 and 70 do the same for the other two. You'll notice that the characters are different from those on line 50. This is one way to get different colors for those characters. This is all explained in the BASIC manual.

Line 80 makes the program wait until you press one of the console keys (START, SELECT, or OPTION).

Line 90 clears the screen back to normal and stops.

Lines 10000 through 10102 were generated by *THE NEXT STEP*. You can figure them out if you want to, but it isn't necessary and it may give you a headache.

7.2 Multicolor Character Animation

This program is a little more complicated, and it should open up worlds of possibility to you. It moves a little red sailboat back and forth over a little line of blue waves. Actually, to demonstrate how to get the fourth color, the little boat is blue from right to left and red from left to right. This is done with a character set of multicolor characters.

The program is on your *THE NEXT STEP* disk as BOAT. The character set is in BOAT,CHA.

Line 10 clears the screen and sets up our initial display list for as BASIC mode 0. It also tells the computer not to display the cursor.

Line 20 calls *THE NEXT STEP* written subroutine that loads the character set from its file and then changes the character set.

Line 30 saves the left margin value then changes it to one that won't interfere with our pretty picture.

Line 40 computes the beginning address for our use of the display list.

Line 50 changes the first line in the display list. This is a little different from all the others. There is some extra information (the screen address) between the first line and the others, so getting to it is a little different. You also have to add 64 to the ANTIC mode number to tell ANTIC to pick up the screen address.

Line 60 changes the rest of the display list lines to the tall multicolor ANTIC mode. Notice that only 12 of these will fit, even though there are 40 characters per line.

Line 70 finishes up the display list. This finish information always immediately follows your last mode line and is always the same.

Line 80 changes the colors to what we want for our picture.

Line 100 displays the line of waves.

Line 110 displays the sailboat for the first time.

Line 120 sets the value for our wait loop so the sailboat won't move too fast.

Line 200 begins the loop that moves the sailboat to the right.

Line 210 displays the sailboat, moved half a character to the right. This makes the movement smoother than moving it one whole character position at a time. Line 220 waits a while.

Line 230 moves the sailboat the other half character to the right and line 240 waits awhile again.

Line 250 completes the loop.

Line 260 checks to see if you are pressing a console key. That's the way to get out of the program smoothly.

Lines 300 through 350 move the sailboat to the left about the same way we moved it to the right.

Line 360 checks for the console again.

Line 400 restores the left margin.

Line 410 clears the screen and fixes the display list we messed up and line 420 stops.

Lines 10000 through 10080 were generated by *THE NEXT STEP* and read the character set from its save file.

7.3 Mixed Mode Graphics

This program illustrates how you really can have lots of graphic modes on the screen at the same time. Its actually a little overly busy to be a good display, but it certainly makes the point. It also uses a few multicolor characters mixed in with the normal ones. (yep, you can do that, too).

The program is on your *THE NEXT STEP* disk as MIXED. Its character set is in MIXED.CHA.

Line 10 calls the subroutine that *THE NEXT STEP* wrote to set up the character set from ROM and DATA. The character set replaces the characters 0-7 with some multicolor characters.

Line 20 clears the screen and gets us a display list and screen area to work with. This particular graphic mode meets our space requirements.

Line 30 turns off the cursor and changes the character set.

Line 40 initializes the FIXPOS subroutine.

Line 50 saves the left margin and changes it out of our way.

Line 100 computes the beginning address for our use of the display list.

Line 110 changes screen line 0 in the display list to BASIC mode 0, normal text.

Line 120 changes screen line 1 to BASIC mode 1, wide text.
Line 130 changes screen line 2 to BASIC mode 2, big text.
Line 140 changes screen line 3 to BASIC mode 0.
Line 150 changes screen lines 4 through 8 to BASIC mode 3, low resolution plotting.
Lines 160 and 170 change screen lines 9 and 18 to BASIC mode 0. Screen lines 10 through 17 are to be BASIC mode 7 (high resolution plot) so we leave them alone.
Line 180 changes line 19 to a single blank line.
Line 190 changes screen lines 20 and 21 to BASIC mode 0.
Line 200 changes screen line 22 to tall multicolor.
Line 210 changes screen line 23 to BASIC mode 0.
Line 220 changes screen line 24 to short multicolor.
Line 230 changes screen line 25 to BASIC mode 2.
Line 240 changes screen line 26 to BASIC mode 1.
Line 250 changes screen line 27 to BASIC mode 0.
Line 260 puts the standard end into the display list.
Line 300 sets us up to display on screen line 0. The variable XXX is a dummy that we won't use. In case you need it for some esoteric purpose the return value of FIXPOS is the address of the screen memory it pointed BASIC at. Because of the way it is implemented, FIXPOS needs its first parameter to be the address of itself. Its second parameter is the mode line number from the display list.
Line 310 gets BASIC pointed to the right place.
Line 320 displays our message. This one will be BASIC mode 0.
Line 330 sets us up to display on screen line 1.
Line 340 points BASIC where we want our message.
Line 350 displays the message. Notice that this one will be BASIC mode 1 and is therefore displayed with a PRINT 6.
Lines 360 through 420 do likewise for screen lines 2 and 3, which are BASIC modes 2 and 0, respectively.
Line 430 sets us up for some plotting on screen lines 4 through 8.

Lines 440 through 490 plot some interlocking diagonal lines in the three colors available. This area of the screen is in BASIC mode 3, low resolution plot.

Lines 500 through 590 are similar to 400 through 490 except the plotting is high resolution BASIC mode 7.

Line 600 sets us up to display on screen lines 18 through 21. They are all the same graphic mode except for screen line 19 which is blank and thus doesn't count.

Line 610 through 640 display our BASIC mode 0 messages.

Line 650 sets us to display on screen line 22.

Line 660 sets our position.

Line 670 displays the characters that are modified to be multicolor. In this case they will come out tall. The characters are first displayed normal, then inverse. This gets us all the colors. Notice that the characters are displayed in pairs to get a display that looks 8 pixels wide for each character.

Lines 680 through 730 are essentially the same as those above. Because of the ANTIC mode number in the display list, the multicolor characters will be short.

Lines 740 through 820 are similar to lines 300 through 380, with the BASIC modes in reverse order.

Lines 900 through 930 wait for a console key, restore the left margin, clear the screen (getting the display list back to normal), and stop.

Once you believe you understand this program and you have it running in front of you, type a BREAK. Then type LIST and see the interesting effect. This is caused by the listing being scrolled through the memory that ANTIC is displaying with our mixed mode display list. A SYSTEM RESET will get things back to normal.

1) In BASIC mode 0, these must be preceded by a escape, CHR\$(27), to be printed as graphics. Also, to type them as a graphic from the keyboard, you must type an escape first. To type them as graphics, after the escape, use the following keys:

| Character | Key |
|-----------|---------------------|
| 91 | escape |
| 92 | control-up-arrow |
| 93 | control-down-arrow |
| 94 | control-left-arrow |
| 95 | control-right-arrow |
| 125 | control-clear |
| 126 | control-backsp |
| 127 | tab |
| 219 | can't be done |
| 220 | shift-delete |
| 221 | shift-insert |
| 222 | control-tab |
| 223 | shift-tab |
| 253 | control-2 |
| 254 | control-delete |
| 255 | control-insert |

2) Inverse value (155) cannot be printed in any mode (it's a newline). If you must display it, you have to POKE it into the screen memory. That's what *THE NEXT STEP* does. Likewise there is no way to type it as a graphic from the keyboard.

3) Can't be printed in BASIC modes 1 or 2 unless you use the plus 32 form to turn off screen clearing.

APPENDIX A

HOW TO TYPE OR DISPLAY ANY CHARACTER

This appendix contains a table like the one on page 55 of the Atari 400/800 BASIC Reference Manual. In this version of the table, the characters that you can't readily determine how to type or display are flagged. The notes then explain how to type or display them. Unless otherwise mentioned, any note that applies to a character applies to the inverse (number plus 128) version of the same character.

| Column 1 | | | | Column 2 | | | | Column 3 | | | | Column 4 | | | |
|----------|-------|----|-----|----------|-----|----|-----|----------|-----|----|-----|----------|-----|-----|-----|
| # | CHR | # | CHR | # | CHR | # | CHR | # | CHR | # | CHR | # | CHR | # | CHR |
| 0 | Space | 16 | 0 | 32 | @ | 48 | P | 64 | | 80 | | 96 | | 112 | p |
| 1 | ! | 17 | 1 | 33 | A | 49 | Q | 65 | | 81 | | 97 | a | 113 | q |
| 2 | " | 18 | 2 | 34 | B | 50 | R | 66 | | 82 | | 98 | b | 114 | r |
| 3 | # | 19 | 3 | 35 | C | 51 | S | 67 | | 83 | | 99 | c | 115 | s |
| 4 | \$ | 20 | 4 | 36 | D | 52 | T | 68 | | 84 | | 100 | d | 116 | t |
| 5 | % | 21 | 5 | 37 | E | 53 | U | 69 | | 85 | | 101 | e | 117 | u |
| 6 | & | 22 | 6 | 38 | F | 54 | V | 70 | | 86 | | 102 | f | 118 | v |
| 7 | ' | 23 | 7 | 39 | G | 55 | W | 71 | | 87 | | 103 | g | 119 | w |
| 8 | (| 24 | 8 | 40 | H | 56 | X | 72 | | 88 | | 104 | h | 120 | x |
| 9 |) | 25 | 9 | 41 | I | 57 | Y | 73 | | 89 | | 105 | i | 121 | y |
| 10 | * | 26 | : | 42 | J | 58 | Z | 74 | | 90 | | 106 | j | 122 | z |
| 11 | + | 27 | ; | 43 | K | 59 | I | 75 | | 91 | | 107 | k | 123 | |
| 12 | , | 28 | < | 44 | L | 60 | \ | 76 | | 92 | | 108 | l | 124 | |
| 13 | - | 29 | = | 45 | M | 61 | | 77 | | 93 | | 109 | m | 125 | |
| 14 | . | 30 | > | 46 | N | 62 | ^ | 78 | | 94 | | 110 | n | 126 | |
| 15 | / | 31 | ? | 47 | O | 63 | - | 79 | | 95 | | 111 | o | 127 | |

—NOTES—

THE NEXT STEP

by The Logic Smiths

Imagine being able to print the letter "A" and get a multi-color space ship. Using *THE NEXT STEP* and a minimum of programming effort, you can do it in no time at all.

THE NEXT STEP contains well-written, easy-to-use documentation with simple BASIC programming examples that show you how *THE NEXT STEP* can help develop colorful graphic displays. Graphics you never thought possible until now.

THE NEXT STEP is a user friendly, menu driven graphics tool kit that allows you to create new character sets or redefine characters to make shapes for use with your basic or machine language programs. *THE NEXT STEP* allows you to save these "new" characters on disk for future use.

THE NEXT STEP is perfect for use on shapes for animation and features a joystick controlled color menu to make your graphics come alive. *THE NEXT STEP* even generates its own code to help you incorporate new characters and shapes into your programs.

THE NEXT STEP allows you to see your shapes as you make them. Now you can determine ahead of time how characters will interact with one another when creating shapes for Character Set or Player-Missile Graphics.

THE NEXT STEP helps you to mix any of ATARI's 14 graphics modes in the same display. *THE NEXT STEP* is a perfect graphics utility for the BASIC or machine language programmer-novice and professional alike.

THE NEXT STEP features full joystick control for ease of use and quick editing.



36575 MUDGE RANCH ROAD • COARSEGOLD, CA 93614

THE NEXT STEP

BY BOB STEWART

© 1981 ON-LINE SYSTEMS



QN-LINE systems



DISKETTE CARE AND HANDLING INFORMATION



Protect
Proteger
保護

Proteger
Schutzen



No
Non
注意

No
Falsch



Insert Carefully
Inserer avec soin
挿入注意

Insertar

Sorgfältig Einsetzen



Never
Jamais
絶対禁止

Nunca
Nie



10 C - 52 C
50 F - 125 F



Never
Jamais
絶対禁止

Nunca
Nie